Exascale Math Position Paper:
Scalable Mesh-Based Simulation Workflows are required for Exascale Computers

Authors: Jeff Banks (LLNL), Lori Diachin (LLNL), Bill Henshaw (LLNL), Mark Shephard (RPI), Onkar Sahni (RPI), Tim Tautges (ANL)

Contact Information: Lori Diachin, LLNL, diachin2@llnl.gov, 925-422-7130

Position Statement: To date, there has been a gap in the exascale discussion regarding issues related to problem set up, complex geometries, mesh generation, and adaptation and/or regeneration. These areas will become critical bottlenecks in next-generation simulations for DOE science, particularly for problems that require high-order problem specification, CAD-based geometries, and dynamic mesh management on extreme scale systems. Due to the increasing need for automation and reliability on exascale machines, all phases of the simulation workflow must be executed efficiently on large numbers of processing cores. Currently, there is some work in this direction, but without significant investment in mathematics research and software development, these issues will significantly impact our ability to perform simulations at extreme scale.

In reviewing the background materials provided on the Exascale Math Working Group website, we find very little reference or discussion of these issues. Most previous discussion has focused on the general issues of exascale computing that are applicable to all areas (e.g., concurrency, fault tolerance, power aware computing, etc). There has also been significant discussion of the opportunities for application advances, but there has been limited discussion of math research challenges in general (scalable solvers being a key exception), and no discussion on the topics listed above in particular.

Position Justification:

High-order problem specification is important to many DOE applications. Many DOE simulation problems are modeled using complex and evolving geometrical domains – e.g. ,accelerators, fusion tokamak devices, nuclear weapons, and energy systems such as wind turbines. Many groups have demonstrated that using high-order discretization methods for these problems can have better algorithmic intensity, and therefore better performance on next generation computers. To maintain high accuracy, such methodologies in turn require the use of high-order geometrical representations. In addition, as resolution needs change (e.g., higher fidelity simulations), as the geometry evolves (e.g., in shape optimization), or as the solution evolves (e.g., mesh adaptation and moving domain problems), the geometry and mesh management tools that support these processes must inevitably become an integral part of the solution process. Thus all operations must be done on the same extreme-scale computers using in-memory linkage of geometry, meshing, simulation, and adaptive control. The use of constant file transfer between these components is unacceptably slow and wasteful of computing resources.

High-order problem specification methods are already a bottleneck on current generation machines. It is important to note that the issues related to high-order problem specification are already problematic for many different problems operating at large scale. In particular, generating a mesh on a complex geometry is often human-labor-intensive and a current bottleneck to efficient simulation. One of the key issues is that CAD representations are not properly managed by current methods. As an exemplar of this problem, it took 6 months of intensive human intervention to resolve issues with imprinting and merging in CAD models to resolve gaps and overlaps in the ITER geometry. An additional significant issue has to do with file I/O in the problem specification stage, both for the mesh and for high-resolution, spatially varying problem input parameters such as material properties. For example, initial meshes with 100's of millions of elements can be generated in parallel for SLAC accelerator geometries in under 10 minutes. However, it takes nearly twice that time to save that mesh to files using parallel I/O. All of these issues will become worse at the exascale when the geometries become more complex and have more detail.

To successfully address scientific challenges at the exascale, we must increase automation and reliability in pre-processing methods, mesh generation, and adaptive control techniques. There has been significant research in these areas already with investments by DOE ASCR and NNSA, resulting in some advances. However, significant improvements are needed in several areas, and there some topics that will arise only at the exascale that we have not yet begun to address.

In particular, here are example areas we believe need improvement to be successful at the exascale:

- *CAD geometry information at exascal*e. Robust methods to manage geometry properly are essential. In particular, how do you handle evolving geometry (e.g., in shape optimization problems, damage modeling, etc.)? How can you deal with geometry from multiple sources including multiple CAD and multiple forms of image data? There has been some initial math research on the errors introduced by geometric approximation in high-order simulations, much more must be done to generalize this to be meaningful.
- *Parallel mesh generation*: To facilitate parallel mesh generation and high order adaptation of both geometry and mesh, it is important to distribute mesh and geometry information in a consistent way. We often distribute the mesh information, but rarely the geometry. There are preliminary methods for parallel mesh generation for some, but not all mesh types that handle complex geometry; more research is needed to ensure efficient, scalable mesh generation, particularly as meshes reach scales of 10^{15} elements or larger. Mesh quality must be addressed at all stages of the solution process and scalable algorithms for high-order (curved) meshes (involving smoothing, swapping, collapsing and refinement) must be developed and refined.
- *Partitioning and load balancing*: As the meshes are generated, they must be partitioned. It is often unclear how to best do this to maximize performance (e.g., by nodes, by elements, or by a combination of criterion). Since these processes become part of the simulation workflow, dynamic load balancing must be fast, scalable and directly consider the needs of each simulation workflow step.

There are additional areas that have not yet been researched such as:

- *Reduced order information or geometric modeling:* To accommodate changes in geometrical fidelity in advanced, adaptive simulations, we must understand how to "coarsen" or reduce geometric representations and input data. Math research is needed to understand when reduced information is adequate, what the accuracy issues are, how do to this in parallel, etc.
- *Dealing with complex multiphysics applications:* New methods for mesh generation/pre-processing/partitioning/interaction will be needed to meet the demands of new multi-physics applications enabled by exascale computing. For example, it is unclear in general how to partition and load balance multi-domain problems or how to maximize memory bandwidth performance and minimize communication costs when multiple methods (e.g., combined mesh and particle based methods) are used in the multi-physics workflow.
- *Efficiencies for Ensemble Simulations*: Extreme scale computer systems will be used for running large suites of problems for uncertainty quantification, shape optimization or design studies, etc. – often simultaneously. How can one leverage mesh and geometry information across runs (e.g., which information to leverage, what optimizations can be made, are their memory compression schemes that can be achieved)?